(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 403 125 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
24.04.1996 Bulletin 1996/17

(51) Int Cl.[6]: **G06F 3/033**

(21) Application number: **90305975.6**

(22) Date of filing: **31.05.1990**

(54) **Zoom mode operations in display apparatus**

Zoommodusbetriebsarten in einem Anzeigegerät

Fonctionnement en mode zoom pour un écran de visualisation

(84) Designated Contracting States:
BE CH DE ES FR GB IT LI NL SE

(30) Priority: **16.06.1989 US 367159**

(43) Date of publication of application:
19.12.1990 Bulletin 1990/51

(73) Proprietor: **International Business Machines Corporation**
Armonk, N.Y. 10504 (US)

(72) Inventor: **Yanker, Peter Cornelius**
Portola Valley, CA 94025 (US)

(74) Representative: **Burt, Roger James, Dr. et al**
IBM United Kingdom Limited
Intellectual Property Department
Hursley Park
Winchester Hampshire SO21 2JN (GB)

(56) References cited:
EP-A- 0 172 433

- **USING PC STORYBOARD INFORMATION UPDATE SN60-1796 November 1985, ARMONK, N.Y. pages 85 - 88;**
- **'Microsoft Windows Paint User s Guide, version 2' 1987 , MICROSOFT CORPORATION**
- **IBM TECHNICAL DISCLOSURE BULLETIN vol. 22, no. 5, October 1979, pages 2103 - 2105; I.D. JUDD ET AL: "Microprocessor-controlled cursors"**
- **PATENT ABSTRACTS OF JAPAN vol. 8, no. 42 (P-256)(1479) 23 February 1984 & JP-A-58 195 237**

EP 0 403 125 B1

Printed by Jouve, 75001 PARIS (FR)

Description

This invention relates generally to zoom mode operations in display systems and to a method of displaying with a zoom function a magnified portion of an image.

In a publication entitled "Using PC Storyboard", Information Update SN60-1796, November 1985, published by the IBM Corporation, Armonk, New York, there is described at pages 85-88 the functionality of a zoom, or image magnification, function for an applications program entitled "PC Storyboard". While providing for individual pel, or pixel, enlargement this prior art zoom function does not centre the zoom function at the cursor position upon the display screen. This prior art zoom function also does not scroll a zoom viewport that displays an unmagnified view of the zoomed image portion.

A publication by Microsoft Corporation entitled "Microsoft Windows Paint User's Guide", Version 2.0, 1987, describes on pages 31-33 a process for "zooming in" on a portion of an image, thereby creating an enlarged image of that portion. The user can turn pixels on and off in the enlarged image, and a small box in the top left corner of the screen shows a view of the area being edited at normal scale. Hence, in this box, the user can see the effects of turning pixels on and off. To scroll the enlarged imaged being worked on, the user drags the pointer displayed in the enlarged image in the direction he/she wants to scroll. The area in the small box moves as the pointer is scrolled, the enlarged image being updated to match the image in the small box once the scrolling action has been completed (see page 32, lines 1-9 and page 66, last paragraph).

According to the present invention, there is provided a method of operating an image editor in response to input from a user of a system having image display means comprising a plurality of physical pels, comprising the steps of: displaying a magnified portion of a first image as a second image comprised of a plurality of logical pels having a size that is a function of a magnification level currently in effect, the second image having a first cursor positioned therein; displaying within the second image a viewport containing a portion of the first image corresponding to the magnified portion, the viewport having a first and a second dimension corresponding to a plurality of physical pels of the first image; the method being characterised by the following combination of steps: (a) responsive to the user repositioning the first cursor such that the first cursor would be located outside of the displayed second image, shifting the second image in increments of logical pels in unison with movement of the first cursor; and (b) providing a second cursor within the viewport having a position that corresponds to the position of the first cursor, and responsive to the second cursor reaching a first edge of the viewport in response to the movement of the first cursor, shifting the viewport by a plurality of physical pels that correspond to the first or the second dimension of the viewport such that the second cursor is positioned substantially adjacent to a second edge of the viewport that is oppositely disposed to the first edge.

In preferred embodiments, the method further comprises the steps, prior to displaying the second image, of: storing in a memory an unmagnified copy of the first image; responsive to an indicator positioned by the user within the first image, determining the indicator position in coordinates associated with the unmagnified copy of the first image; determining the magnification level currently in effect and a required display size of the logical pels; responsive to the determined indicator position and to a predetermined maximum display size, determining coordinates, relative to the unmagnified copy of the first image, of a first window defining the second image, the indicator position being located substantially at a centre of the first window and being represented by the first cursor in the second image.

Such an arrangement can provide the basis for a zoom function that centres the magnified portion of a visual image upon a display cursor and that provides a viewport that scrolls through an image as a cursor is moved.

In one such arrangement disclosed hereinafter, a method of operating an Image Editor having a zoom function is provided wherein other Image Editor actions are invokable and operable in whatever zoom level is currently in effect. The zoom mode is a state of the Image Editor in that invoked actions are performed in a normal 1X pel size image but are displayed to an operator in an enlarged, fat pel, zoom equivalent. when the ZOOM function is invoked a displayed zoom window is automatically centred upon a present position of an indicator, or cursor, as is also a selectively displayable viewport for showing the relative position of the cursor within the 1X image.

It is also possible to determine coordinates within the unmagnified image of a second window for defining a portion of the image to be displayed as an unmagnified image, the indicator position being located substantially at a centre of the second window. The second window is displayed or is not displayed depending upon input from the user. Depending in which area of the images which cursor is moved, both panning and scrolling are enabled and mode details can be switched on the fly.

The present invention will be described further by way of example with reference to an embodiment thereof as illustrated in the accompanying drawings, in which

Fig. 1 is a block diagram of an information processing system embodying the invention;

Fig. 2 shows in detail the organisation of the display screen of Fig. 1;

Fig. 3 illustrates the display screen and shows a magnified portion of an image and a viewport showing an unmagnified portion of the image, both the screen and viewport having an associated cursor; and

Fig. 4 is a flow chart that illustrates a method of initialising the zoom function.

The environment selected for the following disclosure of one form of zoom function according to the present invention is that of an audio/visual application processor implemented on an IBM™ PS/2™ computer system (IBM and PS/2 are trademarks of the IBM Corporation of Armonk, NY). However, it should be realised that the teaching of the invention may be practiced with a number of different types of information processing systems in support-of a number of different types of application programs that provide a visual display.

Referring first to Fig. 1 in conjunction with Fig. 2, there is shown a visual display screen 10 associated with a read/write display memory 1 wherein is stored data to read out and displayed on the screen 10 in a conventional manner. An individual or a group of storage locations within the display memory 1 corresponds to individual pixels or pels of the display screen 10. The display memory 1 is coupled to a central processing unit (CPU) 2, the CPU 2 further being coupled via a bus 3 to a main memory 4 wherein instructions are loaded and stored from a mass storage device 5. Certain of the stored instructions control the CPU 2 to perform an Image Editor function, including the image Zoom or enlargement function of the invention. Other functions associated with the Image Editor include, for example, Text, Line, Sketch, Cut, Paste, Box, and Fill functions. As will be made apparent these other functions may be invoked and executed concurrently with the Zoom function. The other functions operate upon an unmagnified, 1X copy of the image stored within a World Plane (WP) image buffer in memory 4. The enlarged portion of the image that is stored within display memory 1 and displayed on screen 10 is a window or viewport, initially centred upon an image cursor, into the 1X WP image. The enlarged portion of the WP image is displayed with a logical pel size that is a function of the zoom magnification level in effect. An enlarged logical pel is comprised of some number of physical screen pels or pixels.

A data entry device such as a keyboard 6 and a mouse 7 enable a user of the system to interact with the Image Editor in a manner to be described to create, modify and store graphical images. Textual data entered is also temporarily stored in a Text Buffer (TB) within memory 4. The data stored within the display memory 1 may provide information for displaying a monotone or a colour image, depending upon the characteristics of the particular system display screen 10. In the disclosed arrangement, in so far as it is of significance, the display screen 10 displays colour images.

The ZOOM function of the invention provides seven user selectable levels (1X, 2X, 4X, 8X, 16X, 32X and 64X) of image enlargement. A specific zoom level stays in effect until changed in magnification by another ZOOM function or until deactivated. In general, other Image Editor functions are usable within any of the zoom levels.

A ZOOM VIEWPORT function is controlled by a user to switch between displaying a zoom viewport 12 and not displaying the zoom viewport 12. The zoom viewport 12 is a relatively small window within the zoomed display 10 that displays in "normal" size (1X) an area surrounding a cursor 14. An enlarged, fat pel, cursor 16 within the zoomed image corresponds to the cursor 14. The zoom viewport 12 is only displayed when zooming is in effect, that is for image enlargements greater than 1X and when specifically enabled by the ZOOM VIEWPORT function.

Upon selection by a user, the display screen 10 displays the image 18 enlarged in accordance with the particular zoom level in effect. The zoomed area is centred around the cursor 14. The zoom level remains in effect until either changed by a later ZOOM function or until deactivated by an ESC code entered while in an idle state. The entry of the ESC code restores the normal display (zoom level 1X).

Image Editor actions are invokable and operable in whatever zoom level is in effect. For example, CUT and PASTE functions operate at all zoom levels. Further, alphanumeric text may be entered with the TEXT function, an image may be saved to disk or drawing can be accomplished while the ZOOM function is invoked. That is, the zoom mode becomes a state of the Image Editor in that invoked actions are performed upon the unmagnified copy of the image within the WP buffer with a normal 1X pel size but are displayed to the operator in an enlarged, fat pel, equivalent. For example, when entering text while in zoom mode the text is written to the 1X TB in a normal, unmagnified, size as determined by the font and point size in use independently of the zoom level in effect. However, text occurring within the zoom window is displayed enlarged to the current zoom level. Upon a return to normal view (1X) the text appears in the proper size for the standard 1X screen display. When entering text with a large zoom level in effect, such as 64x, the text field may not fit within the screen display. The disclosed arrangement provides for automatically panning the display as needed to keep up with the cursor 16 position.

For example, one Image Editor function is known as FILL. If required by a particular image the FILL function also fills image areas outside of the currently displayed zoom window. Thus, the operation of the FILL function proceeds in a normal (1X) fashion regardless of the zoom level currently in effect. Similarly a PASTE function pastes down a full paste buffer (PB) area, displayed in a properly enlarged format, even though the PB extends beyond the boundaries of the enlarged display. It is noted that write to disk functions such as SAVE and SAVE AND EXIT write the 1X version of the current image to disk even when the ZOOM function is displaying an enlarged view of the image. If desired the enlarged view may be saved to disk using a RESIZE function to generate the enlarged view. The RESIZE function generates an enlarged view of an image at a normal, physical, pel size that becomes a new 1X image. The new 1X image is

then stored using SAVE.

If the ZOOM VIEWPORT function is enabled, the zoom viewport 12 is generated and displayed at a corner of the display 10 whenever a zoom level other than 1X is in effect. The zoom viewport 12 displays in 1X magnification a portion of the image surrounding the cursor 14 location. The zoom viewport 12 thus provides a perspective of the location of cursor 16 within the 1X WP image. The presence of the zoom viewport 14 also indicates to a user that a ZOOM function is in effect. Upon selection by the operator, the ZOOM VIEWPORT function determines either whether the viewport 12 is displayed (ON) during zoom mode or whether it is not displayed (OFF). The option selected remains in effect until changed to the opposite state by another ZOOM VIEWPORT function.

As the operator repositions the cursor 16 within the magnified image window, the cursor 14 moves to reflect the cursor 16 position within the 1X WP image copy. As cursor 16 is panned across the enlarged image the cursor 14 of the viewport 12 moves relative to the image area approximately in unison. The portion of the 1X image contained within viewport 12 changes only when the cursor 14 intersects an edge of the viewport 12. This condition is detected and the coordinates of an adjacent portion of the unmagnified 1X image from the WP buffer are determined and the image displayed. The adjacent portion is sized as a function of the size of the viewport window 12 and the position of the cursor 14. The newly displayed portion of the 1X image has the cursor 14 positioned at the edge of the viewport where it entered, that is opposite the edge that it initially intersected. For example, if cursor 16 movement causes the cursor 14 to intersect, in WP coordinates, the left edge of the viewport 12 of Fig. 3, CPU 2 detects this condition, accesses the 1X image copy and displays the adjacent image portion in viewport 12. In the newly displayed portion the cursor 14 is positioned adjacent to the right edge of the viewport 12. Thus the viewport 12 "pans" in increments of viewport 12 dimensions rather than moving continuously as the cursor 16 and the enlarged view in the zoom window does.

Also, if the cursor 16 intersects an edge of the viewport 12 the viewport 12 is shifted to another position upon the display 10. This enables the zoom area previously under or behind the viewport 12 to be displayed and accessed.

It should be noted that the style of cursor in viewport 12 is the same as the style of cursor specified for use in the Image Editor. At any time, while in zoom mode, a change in cursor style also changes the style of cursor 14. For example, the cursor 14 may have the shape of a cross, a dot or even "invisible".

While in zoom mode, the cursor 16 is a square shaped group of visually distinct pels having a transparent centre region. When the cursor 16 reaches an edge of the display 10 any further attempted cursor movement in the same direction causes the screen 10 to pan in order to accommodate the continuing movement of the cursor 16. Such panning and cursor movement stops when an actual edge of the WP image within memory 2 is reached. For example, when cursoring left and upon reaching the edge of the current display 10 further cursoring left continues, assuming the image edge is not yet reached, and the screen view pans in unison to keep up with the location of the cursor. In this case, image pels on the right edge of the display are pushed off to the right as new pels push on from the left edge to show the newly displayed portion of the image. The WP image is similarly accessed and displayed for cursor 16 movement that occurs when moving right, up or down.

During the display of enlarged zoom views of an image, the cursor 16 moves in increments appropriate for the zoom level. For example, when at a magnification level of 64X, movement typically occurs in increments of a single, enlarged pel. In this regard it is noted that at magnifications above 1X that the cursor 16 is moving in "logical" pels and not in "physical" pels. For example, if zoom level 64X is selected, the display 10 still has a physical resolution of some number of pels, such as 640x480. However, from a logical point of view the image has been enlarged such that only a few "logical" pels of the image are displayed on the screen 10. Each pel is physically sized along both x and y screen coordinates by the CPU 2 such that it encompasses a plurality of pels for display purposes, but each enlarged "fat" pel represents but a single logical pel of the image. From a ZOOM function perspective, cursor 16 movement and the physical size of the cursor 16 both correspond to the logical pel size.

Predetermined cursor direction keys of the keyboard 6 move the cursor 16 accordingly - left, right, up, down, and in the four diagonal directions. CTRL versions of these keys, that is depressing the CONTROL key simultaneously with a cursor direction key, moves the cursor 16 directly to the WP image edges, as is done for normal, non-zoom, cursoring. SHIFT versions of these keys move the cursor 16 in one pel increments, again as for normal cursoring.

Other cursoring conventions are also provided when operating in the zoom mode. ALT versions of predetermined keyboard keys, preferably the numeric keypad cursor keys, move the cursor 16 directly to the edges of the currently displayed image, thereby keeping cursor 16 movement restricted to within the current display. ALT 5 positions the cursor 16 to the centre of the currently displayed image in a manner somewhat analogous to CTRL 5 that moves the cursor to the centre of the WP image.

Further, another special cursoring convention is provided that is enabled by, in the disclosed arrangement, the CTRL Z key. When depressed, this keystroke combination causes the CPU 2 to record the current cursor 16 screen coordinate position and thereafter centre the cursor 16 and the surrounding enlarged portion of the image at the middle of the display screen 10. The effect of this keystroke combination is to centre an enlarged

view of the image around a current location of the cursor 16. If not in zoom mode when this keystroke combination is entered, CPU 2 instead invokes the zoom mode and establishes the display screen 10 accordingly, using the zoom level last selected during a previous zoom mode operation.

If the zoom viewport 12 is enabled during the direct cursoring accomplished in conjunction with the ALT, SHIFT and CTRL keys the zoom viewport 12 is updated in unison with whatever changes occur in the cursor 16 location and screen display.

Depressing another predetermined key, in this case ESC, terminates the zoom mode if no other Image Editor functions are also invoked. Terminating the ZOOM function causes the screen 10 to display a normal 1X image. By example, if drawing a box while in zoom mode, pressing ESC while in the midst of sizing an elastic box outline causes an escape from the BOX function back to an Image Editor idle state. A zoom level presently in effect, such as 32X, continues. If the BOX action is instead terminated normally, subsequentially pressing ESC terminates the ZOOM function. Termination of the ZOOM function causes CPU 2 to save the current zoom state including the current zoom level, for example 32X, and whether the zoom viewport is on or off. When the ZOOM function is again invoked the saved parameters are retrieved and used to initially establish the zoom processor state. Termination of the zoom function also releases the WP buffer area.

Referring to Fig. 4 there is shown in block diagram form the initialisation of the ZOOM function. At block 20 CPU 2 recognises a CTRL Z keystroke from keyboard 6 and invokes the ZOOM function. At block 22 the 1X WP image is established in the WP buffer. In this regard it is noted that if a 1X image already exists the image is copied from display memory 1 into the WP buffer. At block 24 the position of the user positioned cursor is determined within the image, the cursor position being referenced to the coordinates of the WP image. At block 26 a current zoom level is determined and the corresponding expanded logical pel size is calculated. The initial zoom level is set to the level in effect when the ZOOM function was last terminated. At block 28 the coordinates of the zoom window are determined, that is, what portion of the copy of the 1X image in the WP buffer will be displayed in an enlarged format within the zoom window. The coordinates of the zoom window are a function of the cursor position, in that the window is centred on the cursor, the zoom level and the size of the zoom window. Next, at block 30, the coordinates of the viewport 12 are determined. The viewport 12 is nominally defined to have an area equal to approximately one sixteenth of the zoom display window. The zoom display window normally occupies substantially the entire available area of the display screen 10. At block 32 the CPU 2 accesses those 1X pels within the WP buffer that lie within the calculated coordinates of the zoom window. The accessed pels are expanded in accordance with the calculated logical pel size and are written to the display memory 1, thereby creating the expanded image. Also, the 1X pels from the WP buffer are copied to the display memory 1 such that the viewport 12 appears in a corner of the zoom window. Thereafter, the CPU 2 interacts with a user via the keyboard 6 and/or mouse 7 to pan through the image and to invoke other of the image editor functions.

The zoom level may be changed while the ZOOM function is invoked. An operator changes the magnification level via the keyboard 6 and the CPU 2 recalculates the logical pel size and the zoom window coordinates. The zoom window coordinates are a function of the present position of the cursor within the 1X image in that the cursor is by definition taken to be the centre of the window. That is, blocks 24-34 of the flowchart of Fig. 4 are executed.

Claims

1. A method of operating an image editor in response to input from a user of a system having image display means comprising a plurality of physical pels, comprising the steps of:

   displaying a magnified portion of a first image as a second image comprised of a plurality of logical pels having a size that is a function of a magnification level currently in effect, the second image having a first cursor (16) positioned therein;
   displaying within the second image a viewport (12) containing a portion of the first image corresponding to the magnified portion, the viewport having a first and a second dimension corresponding to a plurality of physical pels of the first image;
   the method being characterised by the following combination of steps:

   (a) responsive to the user repositioning the first cursor (16) such that the first cursor would be located outside of the displayed second image, shifting the second image in increments of logical pels in unison with movement of the first cursor (16); and

   (b) providing a second cursor (14) within the viewport (12) having a position that corresponds to the position of the first cursor (16), and responsive to the second cursor (14) reaching a first edge of the viewport (12) in response to the movement of the first cursor (16), shifting the viewport (12) by a plurality of physical pels that correspond to the first or the second dimension of the viewport such that the second cursor (14) is positioned substantially adjacent to a second edge of the viewport that is oppositely disposed

to the first edge.

2. A method as claimed in claim 1, further comprising the steps, prior to displaying the second image, of:

storing in a memory (1) an unmagnified copy of the first image;

responsive to an indicator positioned by the user within the first image, determining the indicator position in coordinates associated with the unmagnified copy of the first image;

determining the magnification level currently in effect and a required display size of the logical pels;

responsive to the determined indicator position and to a predetermined maximum display size, determining coordinates, relative to the unmagnified copy of the first image, of a first window defining the second image, the indicator position being located substantially at a centre of the first window and being represented by the first cursor (16) in the second image.

3. A method as claimed in claim 2, wherein the step of determining the first window coordinates includes a step of determining coordinates, relative to the unmagnified copy of the first image, of the viewport (12) for defining a portion of the first image to be displayed as an unmagnified image, the indicator position being located substantially at a centre of the viewport, and being represented by the second cursor (14) in the viewport (12).

4. A method as claimed in claim 3, wherein the first window has an area equal to approximately a maximum displayable area of the image display means and wherein the viewport (12) has an area of approximately one sixteenth that of the first window.

5. A method as claimed in any of claims 2 to 4, wherein in response to a request by a user to modify the magnification level currently in effect, the method comprises the additional steps of:

determining a required display size of a pel magnified to the modified magnification level; and

responsive to the determined cursor position and to a predetermined maximum display size, determining coordinates, relative to the unmagnified copy of the first image, of the first window, the cursor position being located substantially at the centre of the first window; and

displaying the second image within the first window, the second image being displayed with the magnified pel size at the modified magnification level.

6. A method as claimed in any of claims 2 to 5, wherein in response to input from the user to move the position of the first cursor (16), the method includes a step of repositioning the first cursor (16) in a desired direction to substantially an edge of the first window.

7. A method as claimed in claim 6, wherein in response to a further input from the user to move the position of the first cursor, the method includes a step of repositioning the first cursor to substantially the centre of the first window.

8. A method as claimed in any preceding claim, wherein in response to an input from the user to move the position of the first cursor, the method includes a step of repositioning the first cursor in a desired direction to substantially an edge of the first image.

9. A method as claimed in claim 8, wherein in response to a further input from the user to move the position of the first cursor, the method includes a step of repositioning the first cursor to substantially the centre of the first image.

10. A method as claimed in any preceding claim, wherein the first cursor (16) is displayed within the second image as a visually distinct group of pels having a transparent central portion.

11. A method as claimed in any preceding claim, wherein the second cursor (14) is displayed in the viewport (12) in accordance with a user specified shape.

12. A method as claimed in any preceding claim, wherein in response to a request from the user to modify the image, the method includes the steps of:

modifying an unmagnified copy of the first image; and

for a portion of the first image being displayed as the second image within the first window, displaying a modified second image corresponding to the magnified portion of the modified image within the first window, the magnified portion being displayed with the magnified pel size.

13. A method as claimed in any preceding claim, wherein in response to a request from a user to terminate the display of a magnified portion of the first image, the method includes a step of saving the

magnification level currently in effect such that a subsequent request from a user to re-enter the display of the magnified portion causes the saved magnification level to be the magnification level in effect.

14. A method as claimed in any preceding claim, wherein, responsive to a user causing the first cursor (16) to intersect an edge of the viewport (12), the method includes a step of repositioning the viewport (12) within the second image.

15. A method as claimed in any preceding claim, wherein the step of shifting the viewport (12) in response to the second cursor (14) reaching the first edge of the viewport includes the steps of:

determining coordinates, relative to an unmagnified copy of the first image and to the size of the viewport (12), of a portion of the unmagnified copy of the first image adjacent to the first edge; and

displaying the adjacent portion of the unmagnified copy of the first image within the viewport, the unmagnified portion being displayed with the second cursor (14) positioned substantially adjacent to the second edge of the viewport oppositely disposed to the first edge.

Patentansprüche

1. Ein Verfahren, um einen Bildeditor als Reaktion auf eine Eingabe von einem Benutzer eines Systems mit Bildanzeigemitteln zu betreiben, die eine Vielzahl von physischen Bildelementen enthalten, wobei das Verfahren Schritte enthält, um

einen vergrößerten Teil eines ersten Bildes als zweites Bild anzuzeigen, das eine Vielzahl von logischen Bildelementen enthält, deren Größe abhängig von einem aktuell gültigen Vergrößerungsfaktor ist, wobei das zweite Bild einen ersten Cursor (16) hat, der in diesem positioniert ist;

mit dem zweiten Bild ein Darstellungsfeld (12) anzuzeigen, das einen Teil des ersten Bildes enthält, das dem vergrößerten Teil entspricht und das Darstellungsfeld ein erstes und ein zweites Maß hat, das einer Vielzahl von physischen Bildelementen des ersten Bildes entspricht;

das Verfahren durch die folgende Kombination von Schritten gekennzeichnet wird:

(a) auf die Neupositionierung des ersten Cursors (16) durch den Benutzer zu reagieren, so daß sich der erste Cursor außerhalb des angezeigten zweiten Bildes befindet, wobei das zweite Bild in Schritten von logischen Bildelementen in Übereinstimmung mit der Bewegung des ersten Cursors (16) verschoben wird; und

(b) einen zweiten Cursor (14) innerhalb des Darstellungsfeldes (12) bereitzustellen, das eine Position hat, die der Position des ersten Cursors (16) entspricht, und auf den zweiten Cursor (14) zu reagieren, der einen ersten Rand des Darstellungsfelds (12) als Reaktion auf die Bewegung des ersten Cursors (16) erreicht, wobei das Darstellungsfeld (12) durch eine Vielzahl von physischen Bildelementen verschoben wird, die dem ersten oder zweiten Maß des Darstellungsfelds entsprechen, so daß der zweite Cursor (14) im wesentlichen in der Nähe eines zweiten Rands des Darstellungsfelds positioniert wird, d.h. dem ersten Rand gegenüberliegt.

2. Ein Verfahren wie in Anspruch 1 angemeldet, das außerdem die Schritte enthält, die vor der Anzeige des zweiten Bildes liegen:

Speicherung einer nicht vergrößerten Kopie des ersten Bildes in einem Speicher (1);

Reaktion auf einen Anzeiger, der von dem Benutzer in dem ersten Bild positioniert wird, wobei die Anzeigerposition in Koordinaten bestimmt wird, die zu der nicht vergrößerten Kopie des ersten Bildes gehören;

Festlegung des aktuell gültigen Vergrößerungsfaktors und einer gewünschten Anzeigegröße der logischen Bildelemente;

Reaktion auf die festgelegte Anzeigerposition und auf eine zuvor bestimmte, maximale Anzeigegröße, wobei die Koordinaten bezogen auf die nicht vergrößerte Kopie des ersten Bildes von einem ersten Fenster festgelegt werden, welches das zweite Bild definiert, wobei sich die Anzeigerposition im wesentlichen im Mittelpunkt des ersten Fensters befindet und durch den ersten Cursor (16)in dem zweiten Bild dargestellt wird.

3. Ein Verfahren wie in Anspruch 2 angemeldet, wobei der Schritt zur Festlegung der ersten Fensterkoordinaten einen Schritt enthält, um die Koordinaten im Verhältnis zu der nicht vergrößerten Kopie des ersten Bildes von dem Darstellungsfeld (12) zu bestimmen, um einen Teil des ersten Bildes zu definieren, der als nicht vergrößertes Bild anzuzeigen

ist, wobei sich die Zeigerposition im wesentlichen im Zentrum des Darstellungsfelds befindet und durch den zweiten Cursor (14) in dem Darstellungsfeld (12) dargestellt wird.

4. Ein Verfahren wie in Anspruch 3 angemeldet, wobei das erste Fenster einen Bereich hat, der in etwa gleich einem maximal anzeigbaren Bereich mit Bildanzeigemitteln ist und wobei das Darstellungsfeld (12) einen Bereich von etwa einem Sechzehntel von dem des ersten Fensters hat.

5. Ein Verfahren wie in irgendeinem der Ansprüche 2 bis 4 angemeldet, wobei als Reaktion auf eine Anforderung durch einen Benutzer, den aktuell gültigen Vergrößerungsfaktor zu ändern, das Verfahren zusätzliche Schritte enthält, um

eine geforderte Anzeigegröße eines Bildelements zu bestimmen, das mit dem geänderten Vergrößerungsfaktor vergrößert wurde; und um

auf die festgelegte Cursorposition und auf eine zuvor bestimmte maximale Anzeigegröße zu reagieren, indem die Koordinaten im Verhältnis zu der nicht vergrößerten Kopie des ersten Bildes aus dem ersten Fenster bestimmt werden, wobei sich die Cursorposition im wesentlichen im Zentrum des ersten Fensters befindet; und um

das zweite Bild in dem ersten Fenster anzuzeigen, wobei das zweite Bild mit der vergrößerten Bildelementgröße in dem geänderten Vergrößerungsfaktor angezeigt wird.

6. Ein Verfahren wie in irgendeinem der Ansprüche 2 bis 5 angemeldet, wobei als Reaktion auf die Eingabe des Benutzers, die Position des ersten Cursors (16) zu verschieben, das Verfahren einen Schritt enthält, um den ersten Cursor (16) in einer gewünschten Richtung, im wesentlichen an einem Rand des ersten Fensters, neu zu positionieren.

7. Ein Verfahren wie in Anspruch 6 angemeldet, wobei als Reaktion auf eine weitere Eingabe von dem Benutzer, die Position des ersten Cursors zu verschieben, das Verfahren einen Schritt enthält, den ersten Cursor im wesentlichen im Zentrum des ersten Fensters neu zu positionieren.

8. Ein Verfahren wie in irgendeinem vorhergehenden Anspruch angemeldet, wobei als Reaktion auf eine Eingabe von dem Benutzer, die Position des ersten Cursors zu verschieben, das Verfahren einen Schritt enthält, den ersten Cursor in einer gewünschten Richtung, im wesentlichen an einem Rand des ersten Bildes, neu zu positionieren.

9. Ein Verfahren wie in Anspruch 8 angemeldet, wobei als Reaktion auf eine weitere Eingabe von dem Benutzer, die Position des ersten Cursors zu verschieben, das Verfahren einen Schritt enthält, den ersten Cursor im wesentlichen im Zentrum des ersten Bildes neu zu positionieren.

10. Ein Verfahren wie in irgendeinem vorhergehenden Anspruch angemeldet, wobei der erste Cursor (16) innerhalb des zweiten Bildes als eine visuell deutliche Gruppe mit Bildelementen angezeigt wird, die einen transparenten Mittelteil hat.

11. Ein Verfahren wie in irgendeinem vorhergehenden Anspruch angemeldet, wobei der zweite Cursor (14) im Darstellungsfeld (12) gemäß einer vom Benutzer angegebenen Form angezeigt wird.

12. Ein Verfahren wie in irgendeinem vorhergehenden Anspruch angemeldet, wobei als Reaktion auf eine Aufforderung von dem Benutzer, das Bild zu ändern, das Verfahren Schritte enthält, um

eine nicht vergrößerte Kopie des ersten Bildes zu ändern; und um

für einen Teil des ersten Bildes, der innerhalb des ersten Fensters als zweites Bild angezeigt wird, ein geändertes zweites Bild anzuzeigen, das dem vergrößerten Teil des geänderten Bildes innerhalb des ersten Fensters entspricht, wobei der vergrößerte Teil mit der vergrößerten Bildelementgröße angezeigt wird.

13. Ein Verfahren wie in irgendeinem vorhergehenden Anspruch angemeldet, wobei als Reaktion auf eine Aufforderung von einem Benutzer, die Anzeige eines vergrößerten Teils des ersten Bildes zu beenden, das Verfahren einen Schritt enthält, um den aktuell gültigen Vergrößerungsfaktor zu speichern, so daß eine nachfolgende Aufforderung von einem Benutzer, wieder dorthin zurückzukehren, die Anzeige des vergrößerten Teils den gespeicherten Vergrößerungsfaktor veranlaßt, der gültige Vergrößerungsfaktor zu sein.

14. Ein Verfahren wie in irgendeinem vorhergehenden Anspruch angemeldet, wobei als Reaktion auf einen Benutzer, welcher den ersten Cursor (16) veranlaßt, eine Kante des Darstellungsfelds (12) zu schneiden, das Verfahren einen Schritt zur Neupositionierung des Darstellungsfelds (12) innerhalb des zweiten Bildes enthält.

15. Ein Verfahren wie in irgendeinem vorhergehenden Anspruch angemeldet, wobei der Schritt zum Verschieben des Darstellungsfelds (12) als Reaktion auf den zweiten Cursor (14), der den ersten Rand

des Darstellungsfelds erreicht, Schritte enthält, um

die Koordinaten im Verhältnis zu einer nicht vergrößerten Kopie des ersten Bildes und zur Größe des Darstellungsfelds (12) von einem Teil der nicht vergrößerten Kopie des ersten Bildes zu bestimmen, das an den ersten Rand angrenzt; und um

den angrenzenden Teil der nicht vergrößerten Kopie des ersten Bildes in dem Darstellungsfeld anzuzeigen, wobei der nicht vergrößerte Teil mit dem zweiten Cursor (14) angezeigt wird, der im wesentlichen angrenzend an den zweiten Rand des Darstellungsfelds, der sich gegenüber dem ersten Rand befindet, positioniert wird.

Revendications

1. Un procédé de fonctionnement d'un éditeur d'images, en réponse à une entrée venant d'un utilisateur d'un système comportant des moyens d'affichage d'image comprenant une pluralité d'éléments d'image physiques, comprenant les étapes consistant à :

   afficher une partie agrandie d'une première image à titre de deuxième image constituée d'une pluralité d'éléments d'image logiques d'une taille fonction du niveau d'agrandissement actuellement en application, la deuxième image ayant un premier curseur (16) positionné en elle;

   afficher à l'intérieur de la deuxième image une fenêtre d'observation (12) contenant une partie de la première image correspondant à la partie agrandie, la fenêtre d'observation ayant une première et une deuxième dimensions correspondant à une pluralité d'éléments d'image physiques de la première image;

   le procédé étant caractérisé par la combinaison ci-après des étapes consistant à :

   (a) réagir au positionnement par l'utilisateur du premier curseur (16), de manière que le premier curseur soit placé à l'extérieur de la deuxième image affichée, déplacer la deuxième image par des incréments d'éléments d'image logiques à l'unisson avec le mouvement du premier curseur (16); et

   (b) fournir un deuxième curseur (14) à l'intérieur de la fenêtre d'observation (12), ayant une position correspondant à la position du premier curseur (16) et, en réaction au deuxième curseur (14), atteindre un premier bord de la fenêtre d'observation (12) en réponse au déplacement du premier curseur (16), déplacer la fenêtre d'observation (12) de la valeur d'une pluralité d'éléments d'image physiques correspondant à la première ou à la deuxième dimension de la fenêtre de manière que le deuxième curseur (14) soit positionné sensiblement adjacent au deuxième bord de la fenêtre disposé à l'opposé du premier bord.

2. Un procédé selon la revendication 1, comprenant en outre, avant l'affichage de la deuxième image, consistant à :

   stocker dans une mémoire (1) une copie non agrandie de la première image;

   réagir à un indicateur positionné par l'utilisateur, à l'intérieur de la première image, déterminer la position d'indicateur en termes de coordonnées associées à la copie non agrandie de la première image;

   déterminer le niveau d'agrandissement actuellement en application et la taille d'affichage requise des éléments d'image logiques;

   réagir à la position de l'indicateur déterminée et à une taille d'affichage maximale prédéterminée, déterminer les coordonnées, par rapport à la copie non agrandie de la première image, d'une première fenêtre définissant la deuxième image, la position d'indicateur étant située sensiblement au centre de la première fenêtre et étant représentée par le premier curseur (16) dans la deuxième image.

3. Un procédé selon la revendication 2, dans lequel l'étape consistant à déterminer les coordonnées de première fenêtre comprend une étape de détermination de coordonnées, relative à la copie non agrandie de la première image, de la fenêtre d'observation (12) pour déterminer une partie de la première image à afficher à titre d'image non agrandie, la position d'indicateur étant située sensiblement au centre de la fenêtre d'observation et étant représentée par le deuxième curseur (14) dans la fenêtre d'observation (12).

4. Un procédé selon la revendication 3, dans lequel la première fenêtre a une aire égale à peu près l'aire affichable maximale du moyen d'affichage d'image et dans lequel la fenêtre d'observation (12) a une aire d'à peu près un seizième de celle de la première fenêtre.

5. Un procédé selon l'une quelconque des revendica-

tions 2 à 4, dans lequel en réponse à une requête faite par un utilisateur pour modifier le niveau d'agrandissement appliqué, le procédé comprend les étapes additionnelles consistant à :

déterminer une taille d'affichage requise d'un élément d'image agrandi au niveau d'agrandissement modifié; et réagir à la position de curseur déterminée et à une taille d'affichage maximale prédéterminée, déterminer des coordonnées, par rapport à la copie non agrandie de la première image, de la première fenêtre, la position de curseur étant située sensiblement au centre de la première fenêtre; et

afficher la deuxième image à l'intérieur de la première fenêtre, la deuxième image étant affichée avec la taille d'élément d'image agrandi au niveau d'agrandissement modifié.

6. Un procédé selon l'une quelconque des revendications 2 à 5, dans lequel, en réponse à l'entrée par l'utilisateur pour déplacer la position du premier curseur (16), le procédé comprend une étape de repositionnement du premier curseur (16) dans une direction souhaitée pour se placer sensiblement sur le bord de la première fenêtre.

7. Un procédé selon la revendication 6, dans lequel, en réponse à une entrée supplémentaire par l'utilisateur, visant à déplacer la position du premier curseur, le procédé comprend l'étape de repositionnement du premier curseur pour venir se placer sensiblement au centre de la première fenêtre.

8. Un procédé selon l'une quelconque des revendications précédentes, dans lequel, en réponse à une entrée venant de l'utilisateur, en vue de déplacer la position du premier curseur, le procédé comprend l'étape de repositionnement du premier curseur dans une direction souhaitée afin de se placer sensiblement au bord de la première image.

9. Un procédé selon la revendication 8, dans lequel, en réponse à une entrée supplémentaire par l'utilisateur visant à déplacer la position du premier curseur, le procédé comprend l'étape de repositionnement du premier curseur pour venir le placer sensiblement au centre de la première image.

10. Un procédé selon l'une quelconque des revendications précédentes, dans lequel le premier curseur (16) est affiché à l'intérieur de la deuxième image, à titre de groupe distinct visuellement d'éléments d'image ayant une partie centrale transparente.

11. Un procédé selon l'une quelconque des revendications précédentes, dans lequel le deuxième curseur (14) est affiché dans la fenêtre d'observation (12), selon une forme spécifiée par l'utilisateur.

12. Un procédé selon l'une quelconque des revendications précédentes, dans lequel, en réponse à une requête venant de l'utilisateur visant à modifier l'image, le procédé comprend les étapes consistant à :

modifier une copie non agrandie de la première image; et

pour une partie de la première image affichée à titre de deuxième image à l'intérieur de la première fenêtre, afficher une deuxième image modifiée correspondant à la partie agrandie de l'image modifiée à l'intérieur de la première fenêtre, la partie agrandie étant affichée avec la taille agrandie d'élément d'image.

13. Un procédé selon l'une quelconque des revendications précédentes, dans lequel, en réponse à une requête venant d'un utilisateur pour déterminer l'affichage d'une partie agrandie de la première image, le procédé comprend l'étape de sauvegarde du niveau d'agrandissement actuellement en application, de manière qu'une requête subséquente, venant de l'utilisateur, visant à ré-introduire l'image de la partie agrandie, entraîne l'application comme niveau d'agrandissement du niveau d'agrandissement sauvegardé.

14. Un procédé selon l'une quelconque des revendications précédentes, dans lequel, en réponse à un utilisateur provoquant l'intersection du premier curseur (16) avec un bord de la fenêtre d'observation (12), le procédé comprend l'étape de repositionnement de la fenêtre d'observation (12) à l'intérieur de la deuxième image.

15. Un procédé selon l'une quelconque des revendications précédentes, dans lequel l'étape de déplacement de la fenêtre d'observation (12), en réponse au deuxième curseur (14) pour lui faire atteindre le premier bord de la fenêtre d'observation, comprend les étapes consistant à :

déterminer les coordonnées, relatives à une copie non agrandie de la première image et à la taille de la fenêtre d'observation (12), d'une partie de la copie non agrandie de la première image adjacente au premier bord; et

afficher la partie adjacente de la copie non agrandie de la première image à l'intérieur de la fenêtre d'observation, la partie non agrandie étant affichée avec le deuxième curseur (14) positionné sensiblement adjacent au deuxième

bord de la fenêtre d'observation, qui est disposé à l'opposé du premier bord.

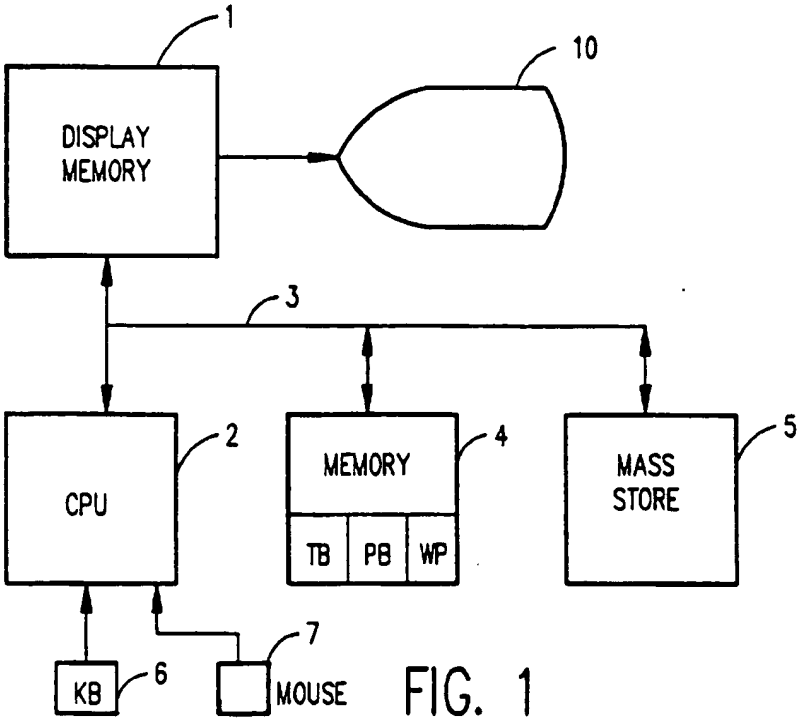
1
10
DISPLAY MEMORY
3
CPU
2
MEMORY
4
TB
PB
WP
MASS STORE
5
KB
6
7
MOUSE

FIG. 1

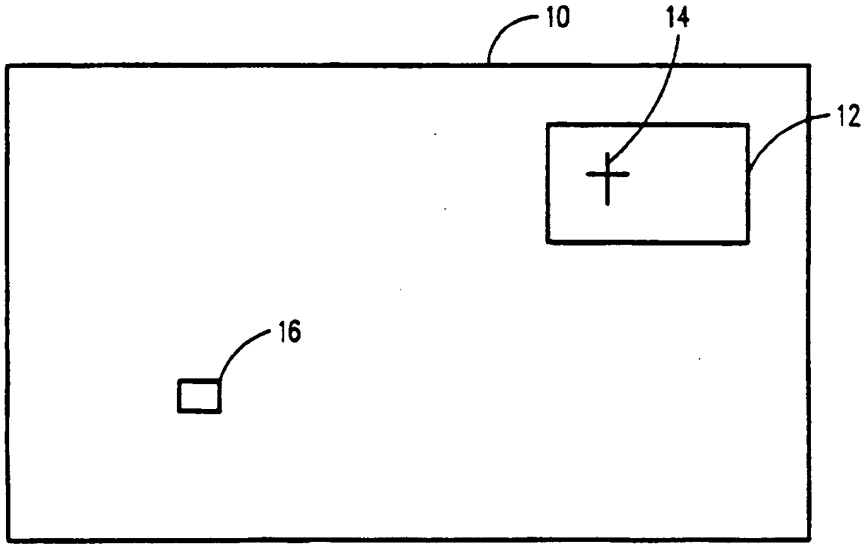
10
14
12
16

FIG. 2

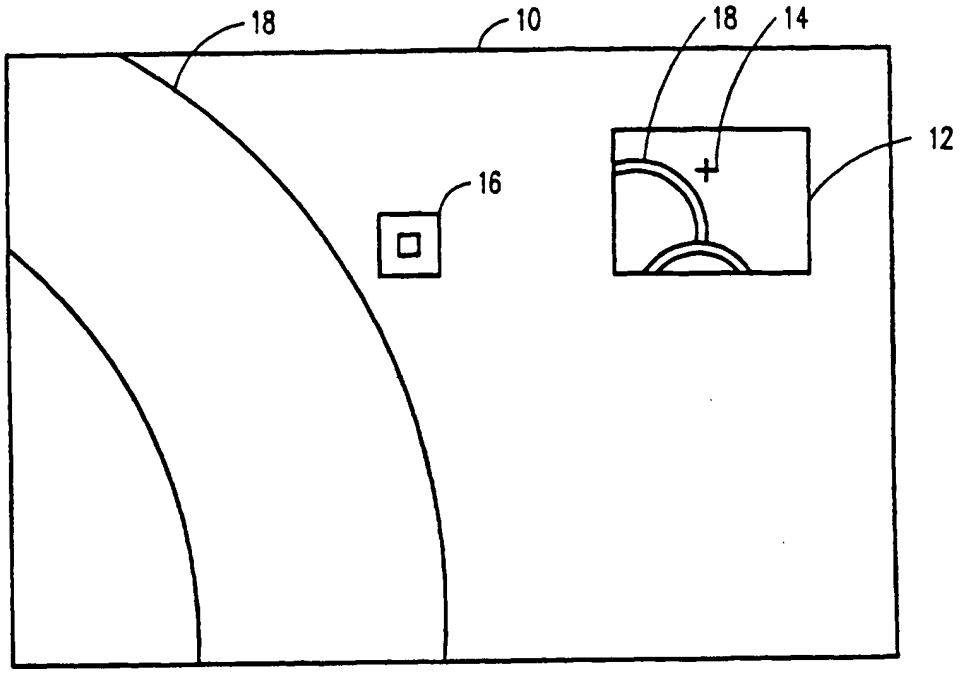

FIG. 3

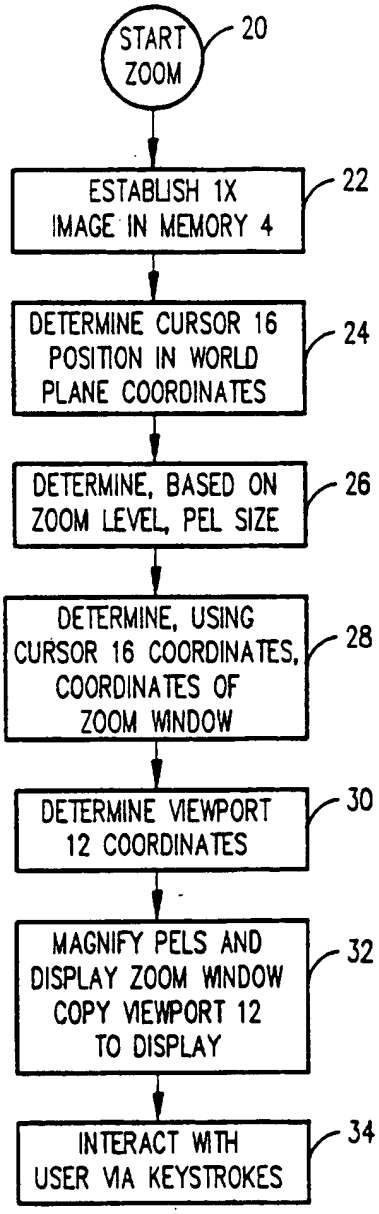
START ZOOM
20
ESTABLISH 1X IMAGE IN MEMORY 4
22
DETERMINE CURSOR 16 POSITION IN WORLD PLANE COORDINATES
24
DETERMINE, BASED ON ZOOM LEVEL, PEL SIZE
26
DETERMINE, USING CURSOR 16 COORDINATES, COORDINATES OF ZOOM WINDOW
28
DETERMINE VIEWPORT 12 COORDINATES
30
MAGNIFY PELS AND DISPLAY ZOOM WINDOW COPY VIEWPORT 12 TO DISPLAY
32
INTERACT WITH USER VIA KEYSTROKES
34

FIG. 4